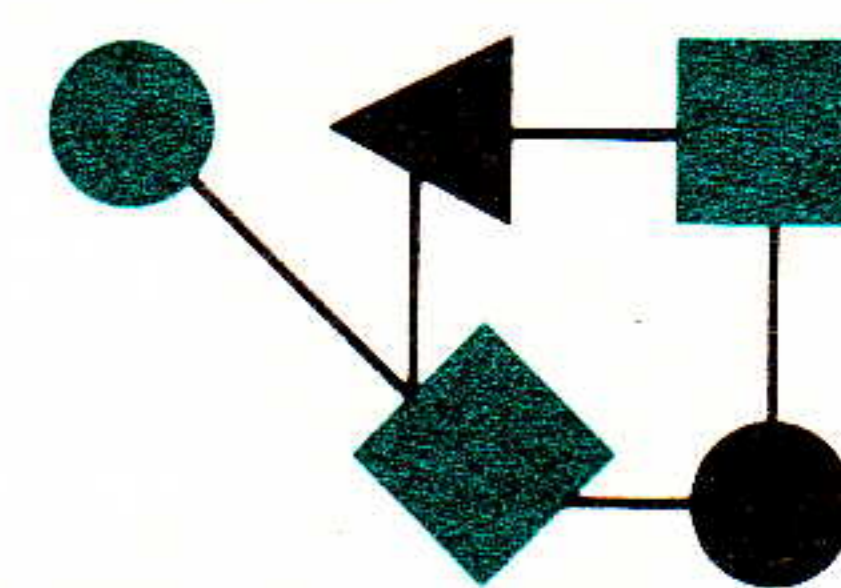


CONNECTIONS



The Interoperability Report

March 1988

Volume 2, No. 3

*ConneXions -
The Interoperability Report
tracks current and emerging
standards and technologies
within the computer and
communications industry.*

In this issue:

Building Distributed Applications in an OSI Framework.....	2
Profile: CSNET.....	8
We're moving!.....	10
Interoperability Survey.....	12
Network Computing Forum..	13
Errata.....	14
Upcoming Events.....	15

ConneXions is published by Advanced
Computing Environments, Inc., 480 San
Antonio Road, Suite 100, Mountain View
CA 94040, USA. Phone: 415-941-3399

©1988
Advanced Computing Environments, Inc.
Quotation with attribution encouraged.

ISSN 0894-5926

From the Editor

This month we feature an article by Marshall Rose of The Wollongong Group, Inc. on how to build distributed applications in an OSI environment. Dr. Rose will cover this material in one of the *TCP/IP OSI/ISO Tutorials* which will take place next month. (See "Upcoming Events", page 15).

In our series profiling existing TCP/IP networks, we look at CSNET. The article is written by the staff of the CSNET Coordination and Information Center at BBN Laboratories.

Last fall we conducted a survey of TCP/IP users. On page 12 we bring you some of the results of this survey.

The Network Computing Forum is a rapidly growing organization with members from industry, academia, and government agencies. The Forum works to reach agreements in the area of network applications. We asked the Forum's director, John Robotham to give us a brief overview of the Forum.

Of our upcoming events (see page 15), I'd particularly like to mention the TCP Performance Seminar which will be held in Monterey, May 9-10, 1988. This seminar will give you an opportunity to learn about important new algorithms which can dramatically improve your TCP's performance.

A *Table of Contents for Volume 1 (1987)* was sent to all subscribers with our February issue. This provides a convenient quick-reference to back issues of *ConneXions*.

We realize that "Advanced Computing Environments" is quite a mouthful and are planning to change our name in the near future. We are announcing a competition to pick our new name. The winner will receive a free conference registration and hotel accommodation for the *3rd TCP/IP Interoperability Conference & Exhibition*, September 26-30, 1988. Entries must be postmarked by March 31st and should be sent to our new address:

Advanced Computing Environments
480 San Antonio Road
Suite 100
Mountain View CA 94040
(415) 941-3399

Building Distributed Applications in an OSI Framework

by Marshall Rose, The Wollongong Group, Inc.

One of the tutorials at the upcoming TCP/IP OSI/ISO tutorial series is entitled "Building Distributed Applications in an OSI Framework." This article serves as an introduction to the topics presented in the tutorial.

Introduction

The popularity of loosely coupled systems has been growing. These systems are often implemented with a "remote procedure call" mechanism, which allows a process on one host to invoke a function on another host. Usually, in response to this invocation, some kind of result, such as a piece of data or an error code, is returned. Because of the close similarity between this interaction and a subroutine call in a programming language, the term RPC, or remote procedure call, is used. One example of an immensely successful system built with the RPC mechanism is the Network File System (NFS). NFS uses an RPC mechanism for storing and retrieving files in addition to manipulating filesystem-related attributes such as ownership information.

Remote operations

In Open Systems Interconnection (OSI), there is a similar concept to RPC, called *remote operations* (RO). This facility is intended to provide the mechanisms for building things as diverse as message handling systems, directory services, network management, and remote database access. Given this broad scope, it's not surprising that the OSI remote operations concept is somewhat general in its nature. As such, the potential payoff is quite high: being able to support a wide range of loosely coupled systems using RO may very well be a key factor in the overall success of OSI. However, this broadened approach is not without its potential pitfalls: by providing such a general service, additional "disciplines" must be used in order to focus RO to be easily used for a given application.

The Applications Cookbook

In response to this dilemma, it was decided to implement an appropriate discipline for The ISO Development Environment (ISODE). Put simply, *The Applications Cookbook* is a set of rules for using remote operations coupled with the necessary local implementation decisions used to encourage a particular style of RO usage. *The Cookbook* centers around four key ideas:

- the use of the C programming language for bindings to RO;
- tools for generating parts of the programs which use RO;
- a run-time environment; and,
- conventions for naming and addressing specific services and entities.

Unfortunately, in its realization, nothing is simply "simple." There is a wide range of trade-offs which must be made with regard to flexibility, ease of use, and performance. As such, to quote Michael A. Padlipsky:

*Optimality differs according to context.*¹

¹ *The Elements of Networking Style and Other Essays and Animadversions on the Art of Intercomputer Networking*, Prentice-Hall, 1985.

A model for Distributed Applications

The approach taken attempts to achieve a balance between flexibility and ease of use. As such, it is optimal only in certain cases; in other cases, it is either too complicated for the distributed applications programmer to use, or not flexible enough to implement the desired RO interactions.

To understand a bit more about the decisions which were made in writing *The Cookbook*, let's briefly take a look at a model for building distributed applications. Although RO was introduced above as the mechanism for realizing loosely coupled systems, there is in fact a more interesting, higher-level, notion involved.

Abstract data types

An *abstract data type* is a concept for describing a data structure which is accessed in a well-defined manner. This has several implications: First, although a given data structure may have a *concrete representation* on a given local system (e.g., "struct" in the C language) its corresponding abstract data type is defined in an implementation-independent fashion, termed its *abstract syntax*.

Second, associated with a data structure is a well-defined set of rules, defined by an *abstract transfer notation*. These rules are used to serialize the abstract syntax and thus generate a data stream unambiguously corresponding to the abstract data type for transmission on the network. There are actually two mappings:

- the data structure is mapped to the abstract syntax for the abstract data type, which is a local issue; and,
- the abstract syntax is mapped to an implementation-independent concrete syntax, which is a serializing activity resulting in a stream of octets.

Given this, we can now view an *operation* as being a primitive action which queries or modifies a given abstract data type in some fashion. For any given abstract data type, there is a well-defined set of operations which may be used on that abstract data type. If we restrict access to the abstract data type to exactly this set of operations, then we have a condition known as *strong typing*.

All of this means that we have introduced an *object model* for distributed applications programming: the use of operations, rather than direct manipulation, provides an important level of indirection: data structures may be accessed without regard to their local implementation!

Operations

Having described the relationship between data structures, abstract data types, and operations, let's consider the generic structure of an operation. An *operation*, in its most primitive form, is a simple request/reply interaction. An operation is *invoked*, and in response, one of three outcomes will be reported:

- if the operation succeeds, then a *result* is returned;
- if the operation fails, then an *error* is returned;
- if the operation was not performed for some reason, then a *rejection* is returned.

continued on next page

Building Distributed Applications (*continued*)

Invocation identifier

Associated with an invocation are several parameters, such as its argument. One other interesting parameter is an *invocation identifier*. This is a "serial number" which may be used to uniquely identify a particular invocation. Since it is possible using RO to have several invocations executing simultaneously, the invocation identifier can be used to correlate returning results with outstanding previous invocations.

It is important to understand the concept of *totality* with respect to operations discussed here: for any given operation, the normal outcome (the result) and all exception outcomes (the errors) are well-defined and mutually exclusive (unambiguously distinguishable).

Underlying Services

What capabilities are available to OSI applications to realize this model? There are three facilities available: *abstract syntax notation one* (ASN.1), which provides a means for describing data structures in a machine-independent fashion; the *remote operations service* (ROS), which provides the rules for requesting actions to be performed elsewhere in the network; and the *binding service*, which provides the mechanisms for establishing communications between two entities. Let's briefly consider each.

Abstract Syntax Notation One (ASN.1)

ASN.1 is really two ideas, both of which were introduced in our discussion of abstract data types. The first notion is that of abstract syntax, which defines data structures in a machine-independent fashion. This is realized by using the formal ASN.1 language to describe the data structures corresponding to the abstract data types. The second notion is that of abstract transfer, which says how to serialize data structures defined by an abstract syntax for transmission on the network.

One might think of abstract syntax as defining the vocabulary of some language. That is, it says what words there are and what meaning or interpretation is assigned to each word. In contrast, one might think of abstract transfer as defining the pronunciation of the words in the language. (This analogy is due, in part, to Glenn Trewitt of Stanford University.)

Remote Operations Service (ROS)

The ROS provides facilities needed to send and receive RO information over the network. By itself, the ROS is very simple: it only needs to package the parameters associated with an invocation, result, error, or rejection, and give them to the ISO presentation service for transmission on the network.

The presentation service will serialize the ROS objects into a stream of octets and in turn ask the ISO session service to transmit the resulting octet stream. There is an important implication of this which has a profound impact on the use of remote operations in OSI. Currently, the ROS is defined only for mappings to connection-oriented services. That is, when an association is bound between two entities (discussed below), that association is ultimately realized via a connection-oriented transport protocol. The ROS protocol does not use a datagram service (such as the DoD UDP). Work is proceeding in the standards bodies to permit this functionality.

Binding service

Thus far, we have implicitly assumed some communications relationship (i.e., a *binding*) between the entity invoking operations, and the entity performing them. It is necessary to formalize these notions.

An *association* is a binding between two entities, which are referred to as the *initiator* and the *responder*. Although this sounds straight-forward, there is actually one level of indirection present: the initiator does not directly select a responder for binding, but rather selects a *service* that it wishes to use. For the selected service a responder is located and the binding established. Thus, the binding process is actually two-step:

- the initiator determines which service it requires and asks to have this service mapped onto the entities available on the network; and then,
- based on the initiator's communications requirements, an association will be bound to one of those entities, which becomes the responder.

In OSI, the *directory service* performs the first mapping for the user. It is the responsibility of the initiator's local transport provider to realize the second mapping (which is not as hard as it sounds).

Static facilities

The definition of any loosely coupled system which is to be built using RO usually starts with a formal definition of the remote operations. This is called an RO specification. Although space limitations prevent us from giving an example here, a RO specification can be described as using the data structure description language aspects of ASN.1 to enumerate the complete set of operations, errors, and abstract data types which are used by the system. *The Cookbook* provides a toolkit which starts with a RO specification and successively refines it into a concrete realization for a computer system running ISODE. Rather than describe the toolkit in any detail, let's take only a cursory glance and consult Figure 1 instead.

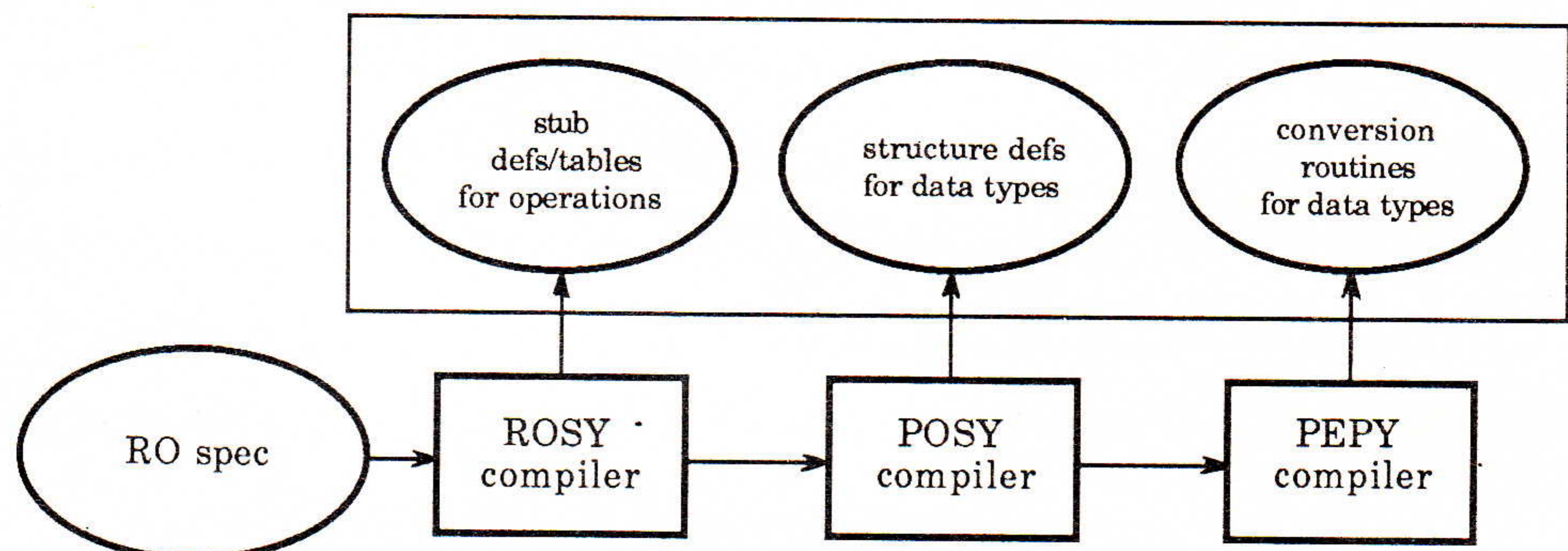


Figure 1: Static Facilities

First, the RO specification is run through a remote operations stub-generator called *rosy*. A "stub" is a subroutine which packages its arguments for use with the remote operations service. The purpose of the stub is to try to hide the network from the distributed applications programmer.

Building Distributed Applications (continued)

Next, the data types which are used by the system are run through another program, called *posy*, which generates a concrete representation for each data type. In our context, the C programming language is used for bindings to RO: the stubs and concrete representations are realized using C. That is, the ASN.1 data types are translated into corresponding C language “structs”.

Finally, we need a way to map between the concrete representation and the abstract syntax. This is what the third compiler in Figure 1, *pepy*, does for us: it defines this implementation-dependent mapping.

The reason these facilities are called “static” is because they are derived prior to execution of the system. Note that the process (using *The Cookbook*) is entirely *automatic*: starting from the RO spec, the stubs, structures, and mappings are all generated without programmer intervention.

Dynamic facilities

In addition to the static facilities, we also need a set of support routines which are used by the system as it executes. These are the so-called “dynamic” facilities. In OSI nomenclature, we use the term *application-specific element* (ASE) to denote the part of the entity which is specific to the application we wish to implement. We note that the application already has several services available to it: an Association Control Service Element (ACSE), which is responsible for establishing and terminating the associations used by the entity; a Directory Services Element (DSE), which is responsible for mapping the services required by the system onto the entities available on the network; and, a Remote Operations Service Element (ROSE), which is responsible for the packaging of invocations and their outcomes.

The question now becomes: between the ASE and these three service elements, what facilities are needed? Rather than describing them in any detail, let's again take a cursory glance and consult Figure 2.

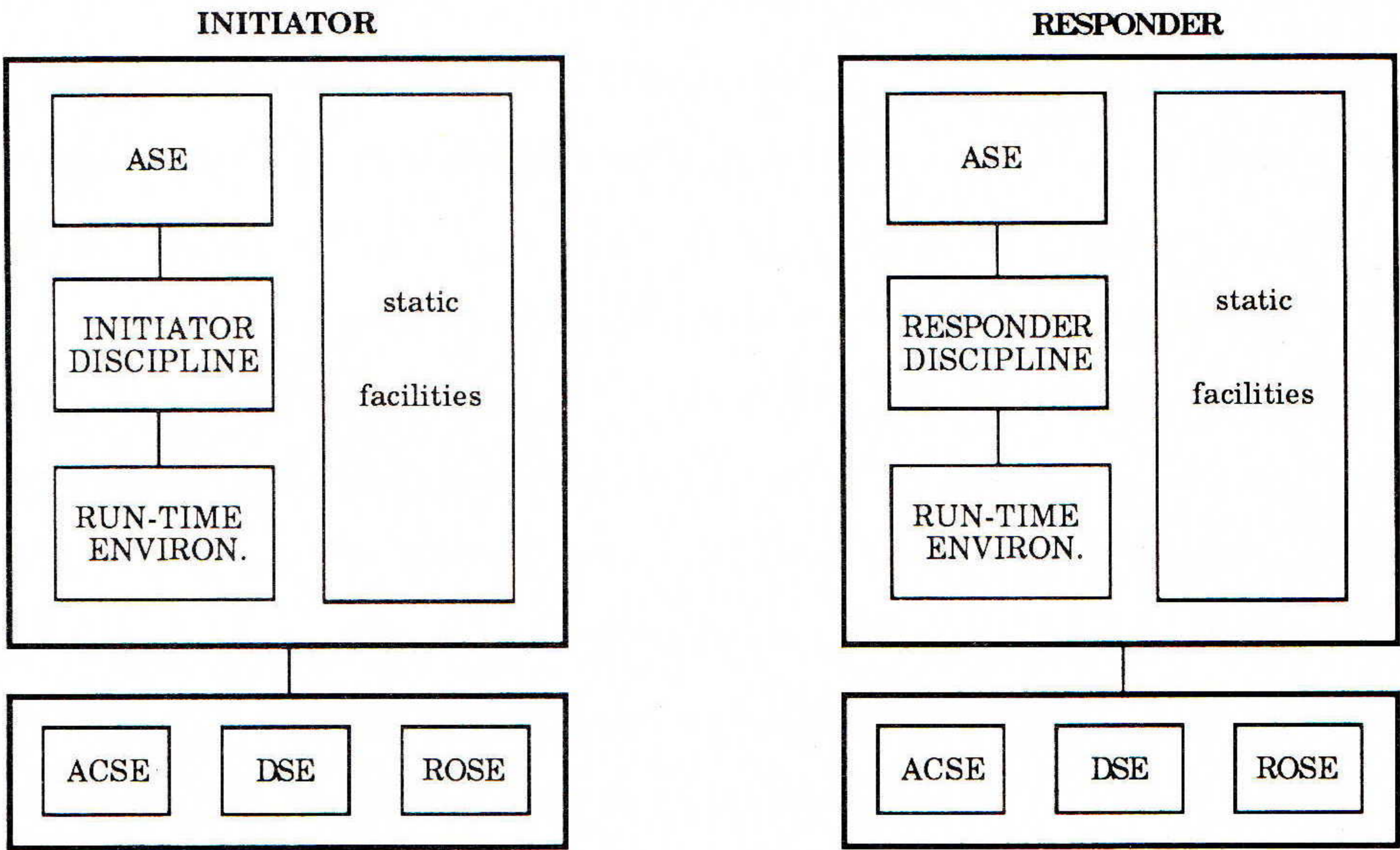


Figure 2: Dynamic Facilities

For the initiator, each stub calls the run-time system to map its C structure arguments into the corresponding abstract syntax and asks to have this given to the ROS for delivery. The stub will then wait for a result, error, or rejection to be returned. It will then map the outcome into the corresponding C structure and return this to the caller.

For the responder, we need to be able to register the operations we will support and then provide a callback mechanism. The "callback" is used when an operation invocation arrives, the argument to the operation is mapped into the corresponding C structure and then given to the callback routine. Then the routine attempts to perform the operation, and returns a result, error, or rejection. Upon doing so, the run-time environment maps the C structures returned into the corresponding abstract syntax and gives them to the ROS for delivery.

The dynamic facilities are table-driven in the sense that they are generic to any application written using *The Cookbook*. The information generated by the static facilities are used to provide the run-time environment with the information needed to perform the mappings accordingly.

Conclusions

As presented in an OSI framework, the concept of remote operations has promise. But, the software infrastructure needed to realize an implementation on a local system is not trivial. In presenting *The Applications Cookbook*, one design for this infrastructure has been described. Obviously other approaches are possible. Indeed, by optimizing for different environments or specific systems, another approach might be vastly superior. Experience will tell. Thus far use of *The Cookbook* suggests that it is a reasonable general facility upon which other, more focused, environments can be based.

Acknowledgements

The ideas presented in *The Applications Cookbook* are due to discussions between the author, Steve E. Kille, of the Computer Science Department, University College London, and Julian P. Onions of the Computer Science Department, University of Nottingham.

[Ed. See also "ISODE: Horizontal Integration in Networking", *ConneXions*, Volume 1, No. 1, May 1987].

MARSHALL T. ROSE is a Principal Software Engineer at The Wollongong Group, Inc. He is the principal implementor of the ISO Development Environment (ISODE), an openly available implementation of the upper layers of the ISO protocol stack. He was co-author of RFC 1006 (ISO Transport Services on top of the TCP), and was a member of the IFIP working group committee whose efforts led to RFC 987 (Mapping between X.400 and RFC 822). He is currently an advisor to the National Science Foundation, serving on its Network Technical Advisory Group. He is also an adjunct Assistant Professor at the University of Delaware. Rose received the Ph.D. degree in Information and Computer Science from the University of California, Irvine, in 1984.

Profile: CSNET - The Computer + Science Network

by CSNET Coordination and Information Center staff,
BBN Laboratories, Inc.

History CSNET, the Computer+Science Network, was established in 1981 with initial funding from the National Science Foundation. The goal of CSNET was to increase the quantity and quality of research in computer science and to bring services such as remote login, file transfer, and electronic mail delivery to the entire computer science research community. The founders of CSNET had observed that a split was developing between scientists with Arpanet access -- the Arpanet was established in 1969 -- and those without. CSNET was designed to redress this imbalance.

Membership Today, CSNET is open to all organizations, including industrial, academic, government, and nonprofit institutions, that conduct or support research or advanced development in science or engineering. The network is used to support a remarkable variety of projects - in computer science, electrical engineering, industrial engineering, industrial automation, artificial intelligence, biology, mathematics, and chemistry. CSNET has nearly 200 members, including Apple Computer, Digital Equipment Corporation, DuPont, General Motors, Hewlett Packard, IBM, Intel, Semiconductor Research Corporation, NASA, the National Science Foundation, and many colleges and universities. Foreign organizations and academic research networks from twelve countries are also members or affiliates of CSNET.

Connection options CSNET provides a variety of access paths for users who wish to pay for different levels of network service. Typically, CSNET members connect to the network by running either PhoneNet or X25Net software.

PhoneNet PhoneNet provides a store-and-forward electronic mail service that allows CSNET member organizations to exchange messages with each other and with major mail networks, including the DARPA/NSF Internet (the successor to the Arpanet), through a central host called relay.cs.net. PhoneNet uses dial-up telephone connections at 1200 or 2400 baud. Messages on PhoneNet, and on the other components of CSNET, follow the Internet standard for the format of text messages (RFC 822 and related specifications).

X.25 X25Net is a full-service Internet-connected network that uses TCP/IP protocols over the public X.25 network Telenet. It provides file transfer, remote login, and immediate electronic mail service between X25Net hosts. With prior approval from DARPA, X25Net hosts may also connect directly to other hosts on the Internet. Many international X25Net members connect to their local public data network (PDN) using the PAD protocol. By way of the PDNs, PAD sites can transfer mail from around the world to CSNET.

Dial-up IP The CSNET technical staff recently completed a version of Dial-up IP software for BSD UNIX and is beta testing it with selected CSNET sites. Dial-up IP software allows sites using the switched telephone network to send IP packets through relay.cs.net to the Internet. The full DoD protocol suite is supported including: TCP, UDP, Telnet, FTP, SMTP, rcp, and rlogin.

Custom connections

In addition to PhoneNet and X25Net, CSNET can provide custom connections for sites with unusual needs. The Massachusetts Microelectronics Center (M2C) was CSNET's first "cluster" site; participating universities connect to M2C via leased telephone lines, and M2C is linked to CSNET by a leased telephone line running TCP/IP. The result is a 9600-baud synchronous connection that allows M2C members to reach the Internet via M2C and relay.cs.net.

A second cluster, using the Cypress packet-switched technology developed at Purdue University, has been established in the San Francisco Bay area. Participants are connected to a hub machine at Digital Equipment Corporation's Western Research Laboratories (DECWRL) at a speed of 9.6 kbps. A leased digital circuit at 56 kbps links DECWRL and Purdue, where operations are centered. Purdue will continue to be the site of the gateway between the Cypress network and the Internet, as it has since 1985.

In the future, CSNET hopes to enhance the connection options available to its members by establishing cluster sites around the U.S.

CSNET Services

Staff at the Coordination and Information Center (CIC), located at BBN Laboratories, Incorporated, in Cambridge, MA, provide technical, operational, administrative, and information services for CSNET. A popular service provided by the CIC is the CSNET hotline, which provides callers with help and information about the network twenty-four hours a day.

Domain servers

CSNET provides its members with valuable administrative services. For instance, primary and backup domain services ensure that electronic mail is consistently delivered to its proper destination. The CSNET staff has been working with the Internet for many years. Their experience with network organizations helps to smooth the introduction of a new member into the research networking community.

Info server

The CIC also offers information services that can be accessed on line. The "Info Server" is an automated, mail-based program that accepts specially formatted mail from users and automatically sends the requested documents by return mail. The Info Server contains information about how to form Internet addresses, a bibliography of articles and reports about CSNET, selected reports in the Request for Comments (RFC) series published by the DDN Network Information Center, public domain software, and information on network hosts and gateways.

User Name server

The CSNET User Name Server is a database of CSNET users and sites that is maintained on the CSNET Service Host (sh.cs.net). All authorized users at CSNET sites may register in the User Name Server. In addition, every CSNET member site has a site entry containing information about the location, host names, and liaisons for the site.

CSNET and other national networks

CSNET has been designated a mid-level network by the National Science Foundation. In effect, this recognizes CSNET as a provider of access to NSFNET, a national high-speed backbone network linking six supercomputer centers, from Princeton to San Diego.

Profile: CSNET *(continued)*

A number of feeder networks, referred to as "regional" or "mid-level" networks by NSF, have been established to link end-users at universities and other organizations to the backbone. TCP/IP is the protocol suite used on NSFNET, and IP-based types of CSNET service are particularly appropriate for institutions that want access to NSFNET.

CSNET is also a charter member of the Federation of American Research Networks, which was formed in 1987. The purpose of the Federation is to represent the combined interests of its members by promoting an integrated high-speed national data communications network for education and research, and addressing the issues that affect providers of mid-level network services. Members expect that the Federation will be able to represent them in relations with NSF, other funding agencies, and standards organizations.

Internics Working Group

Finally, CSNET is a member of the Internics Working Group of the Internet Engineering Task Force. Internics, which consists of representatives from the network information centers (NICs) of various national networks, was founded to coordinate services and foster cooperation among NICs. The most recent meeting focused on solving the global problems of user name servers. CSNET staff were part of the Internics group that prepared a "white paper" examining the types of information that a minimal implementation of a global user name server should contain, the issues that should be considered in establishing such a database (for example, architectures, protocols governing access, replication, and registration controls), and the extent to which the data in the database should be distributed throughout the Internet.

The map on the facing page shows CSNET as of Dec. 15, 1987 ----->

For more information about CSNET call the CIC at (617) 873-2777 or send an electronic mail message to cic@sh.cs.net.

We're moving!

Advanced Computing Environments is moving to new offices. From March 1, 1988 we will be located at:

480 San Antonio Road
Suite 100
Mountain View, CA 94040

Telephone: (415) 941-3399



TCP/IP Interoperability Survey results

In the fall of 1987, we conducted a survey of current TCP/IP users. We asked about 2000 users to list their configurations and note any particular problems and/or missing features. (The underlying assumption was that most TCP/IP products do in fact interoperate as advertized). Only about 100 forms were returned to us completed, so it is difficult to draw any statistical conclusions or plot graphs. Instead we will list some of the dominant trends.

Multi-vendor environments

Most sites have true multi-vendor environments, with equipment from 5 to 10 different manufacturers. (One user listed a total of 18 products from a dozen different vendors). In general, users are impressed with TCP/IP's ability to provide interoperability.

Incomplete implementations

A major complaint for many systems is the lack of completeness of a particular implementation. For example, many PC based implementations do not have SMTP and only support TFTP rather than "real" FTP in some cases. The lack of completeness also takes another form: Vendors are not always following the "current spec" in their implementation. Some vendors are still offering the Name Protocol (IEN-116), rather than the Domain Name System. In their defense, it should be pointed out that some of these protocols are under constant development, and unless you are intimately connected with the "protocol gurus" it is somewhat difficult to track this moving target. Of the "modern features", support for Subnets (RFC 950) is the one most lacking in current implementations. Subnets are required for most of today's complex internets.

Integration with existing software

If TCP/IP (and its associated application protocols) is purchased as an add-on package for a given computer system, the integration of TCP/IP with existing software is often troublesome. For example, no one is particularly happy with any existing solution to integrating TCP/IP-based mail to the native VMS mailer.

Berkeley children

Since many TCP/IP implementations are derived from the UNIX 4BSD distribution, some of the early "known" BSD bugs are still around, examples include "old style broadcast addressing" and the CR-LF mapping in Telnet.

Documentation and software management

Several of the users surveyed complained about poor documentation supplied with their TCP/IP implementation. In many cases it takes a good "hacker" to tweak the system, working with configuration files and installing patches. As with other software, "version management" can often be a problem, getting the latest "fix" from your vendor is sometimes a painful process, when in fact it should be as simple as a subscription service.

Reliability

Reliability of some implementations is not always great. To quote one of the users surveyed: "The biggest problem with this piece of trash is not in the interoperability but with the robustness of the software - it is constantly crashing (4 to 7 times a day)."

TCP/IP Reports

Advanced Computing Environments together with Infonetics are conducting more comprehensive market and user studies and will be publishing two reports in the near future. One is entitled *TCP/IP: The User Perspective 1988*, and the other is called *TCP/IP: The Market View 1987-1990*. For more information call us at 415-941-3399 or Infonetics, Inc at 408-746-2500.

The Network Computing Forum

by John S. Robotham, Apollo Computer Inc.

- Membership** The next meeting of the Network Computing Forum will be held May 24-27 in Ann Arbor, Michigan. This will be the third meeting of the Forum, and will focus on the role of the Forum as a catalyst for change in the industry. The Forum is an industry group chartered to lead the way for rapid adoption of multi-vendor network computing concepts and technologies. Forum meetings allow technical representatives from end user organizations, academic centers, government agencies, software suppliers and hardware vendors to work together on common issues in a open, informal atmosphere. The May 1988 meeting is being hosted by the University of Michigan, at the Michigan League conference center.
- Agreements** There is a growing recognition of Network Computing as an important next generation technology. The first group of applications, system services and tools to support Network Computing are beginning to emerge from both industry and the research community. The Forum is a response to the need for a "critical mass" of hardware vendors, software vendors and end users agreeing on some basic mechanisms to implement Network Computing applications and system environments. The Forum will allow a range of ideas, architectures, and viewpoints to be presented and reviewed. The goal of the Forum is to adopt some common approaches to the problems and opportunities of making this next generation a practical reality.
- Rapid growth** From the outset, the response to the Forum has been outstanding. In March 1987, 31 corporations and academic organizations joined together as charter members of the Forum. Since then, Forum membership has grown to over 100 organizations, with over 200 representatives attending the last meeting in November 1987. The Network Computing Forum is associated with the Corporation for Open Systems (COS) through the COS Affiliate Associate program.
- Forum organization** The Forum is organized into working groups that focus on specific topics of interest to member organizations. Current working groups are:
- User/Application Working Groups:
 - User Requirements
 - Organizational Issues
 - Application Partitioning
 - Tools/Technology Working Groups:
 - Core Network Services
 - User Interface
 - Network Computing on IBM Mainframes
 - Administration/Management Working Groups:
 - Formal Models and Methods for Network Administration
 - Software Licensing
 - Network Environment Management

continued on next page

The Network Computing Forum *(continued)*

The Forum's guiding principles are to be open, informal, cooperative, technical and pragmatic. The diversity of membership - the mixing of hardware vendors, software suppliers, and end users of various kinds - is a key strength of the Forum. As an informal organization, emphasis is placed on exchange of ideas and reaching consensus on common technical approaches to network computing. Forum position papers will be submitted to the appropriate standards organizations, but a formal standards making process is *not* part of the Forum's mission. The need to solve some common, industry-wide technical problems is what brings organizations into the Forum. A cooperative spirit is the hallmark of the group, as is its technical and pragmatic orientation. The Forum concentrates on what can be done in a 12 to 18 month time frame to resolve practical issues in multi-vendor network computing.

Sponsorship

Apollo Computer is the Forum's founding sponsor, providing management and logistical assistance on a voluntary basis. Other Forum members are also contributing their time, expertise and (in some cases) financial support to make the Forum a successful organization. For example, Sun Microsystems acted as sponsor of the November 1987 Forum meeting, and the University of Michigan is hosting the May 1988 meeting. The Forum Steering committee consists of twelve volunteers from the executive committee, along with the Forum's managing director. For more information on the Network Computing Forum contact:

John S. Robotham
Forum Managing Director
c/o Apollo Computer Inc.
330 Billerica Road
Chelmsford, MA 01824
(617) 256-6600 x7875

Errata

In our January 1988 issue Figure 2 on page 4 is incorrect. Both boxes on the TCP/IP side should contain RFC 1006. The correct diagram appears below. Also, in the February 1988 issue, on page 4, line 6 from the top should read: "The common solution to this problem is to include an **NSAP selector field** in the destination address".

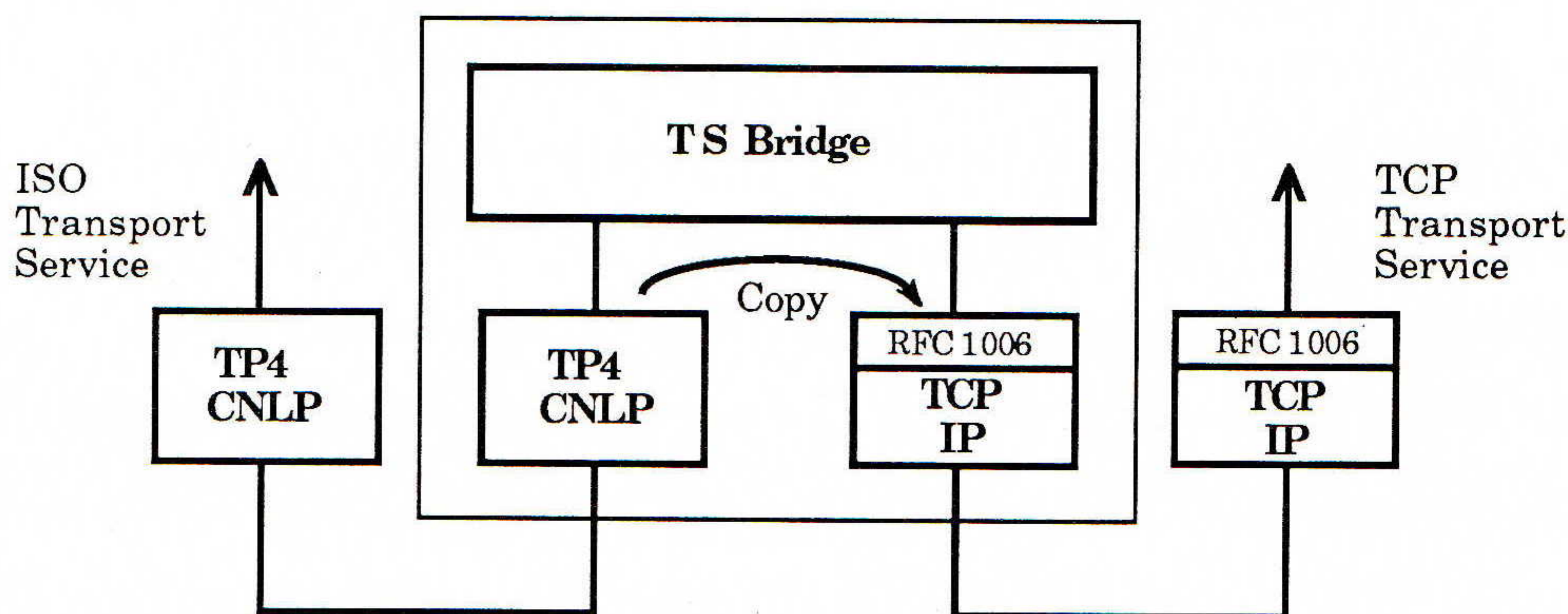


Figure 2: Transport Level Bridge

Upcoming Events

TCP/IP OSI/ISO Tutorials

Advanced Computing Environments will offer 9 *TCP/IP OSI/ISO Tutorials*, April 25 - 27, 1988 at the Hyatt Regency Crystal City in Arlington, Virginia. The tutorials are:

- *Introduction to TCP/IP* (1 day)
- *TCP/IP In-Depth* (2 days)
- *Network Security* (1 day)
- *OSI Reference Model and Protocols* (1 day)
- *Building Distributed Applications in an OSI Framework* (1 day)
- *Local Area Networks* (2 days)
- *Berkeley UNIX Networking* (2 days)
- *TCP/IP for the VM Systems Programmer* (2 days)
- *Microcomputer Networking with TCP/IP* (1 day)

TCP Performance Seminar

We will also sponsor a *TCP Performance Seminar*, May 9-10, 1988 in Monterey, California. Speakers are Van Jacobson and Mike Karels of UC Berkeley. This seminar will be tutorial in nature, presenting new algorithms designed and implemented by the instructors. These algorithms allow a TCP implementation to dynamically determine the proper window size and output rate for optimal performance and minimal congestion. Use of these techniques dramatically increases TCP throughput on slow and/or lossy networks with limited buffering. Other current changes improve performance on fast LANs. The new algorithms discussed will include "slow-start", a technique for determining the buffering capacity of the network and avoiding packet loss in intermediate gateways by avoiding data bursts. Other techniques to be covered include better round-trip-time estimation, retransmission strategies based on timing variance, and lost-packet detection and retransmission without severe performance degradation. Combination of these changes dramatically reduces both spurious retransmission and delays due to lost data. These techniques have been implemented in UNIX 4.3BSD TCP. Additionally, the seminar will describe algorithms for small-packet avoidance, silly-window syndrome avoidance and related changes in the released version of 4.3BSD. Other proposals for congestion control such as Source Quench Introduced Delay (SQuID) will be discussed.

Call us at 415-941-3399 for more information on these events.

IFIP 6.5 Message Handling Conference

The 1988 *IFIP 6.5 International Working Conference on Message Handling and Distributed Application Systems*, will be held October 10 - 12 at the Red Lion Inn in Costa Mesa, California. It is sponsored by IFIP TC 6 and the University of California at Irvine. The purpose of the conference is to provide an international forum for the exchange of information on the technical, economic, social, and political impacts and experiences with computer messaging and distributed application infrastructures in the automated workplace environment. For more information contact:

Einar Stefferud, Conference Chair
Network Management Associates
17301 Drey Lane
Huntington Beach, CA 92647-5615
(714) 842-3711

E-mail: IFIP65@ICS.UCI.EDU
IFIP65@UCI.BITNET
IFIP65@CERNVAX

(Internet)
(BITNET)
(In Europe)

CONNEXIONS

480 San Antonio Road
Suite 100
Mountain View, CA 94040

FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1

CONNEXIONS

PUBLISHER Daniel C. Lynch

EDITOR Ole J. Jacobsen

EDITORIAL ADVISORY BOARD Dr. Vinton G. Cerf, Vice President, National Research Initiatives.

Dr. David D. Clark, The Internet Architect, Massachusetts Institute of Technology.

Dr. David L. Mills, NSFnet Technical Advisor; Professor, University of Delaware.

Dr. Jonathan B. Postel, Assistant Internet Architect, Internet Activities Board; Associate Director, University of Southern California Information Sciences Institute.

Subscribe to CONNEXIONS

U.S./Canada \$100. for 12 issues/year
International \$ 50. additional per year

Name _____ Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Country _____ Telephone () _____

☐ Check enclosed (in U.S. dollars made payable to CONNEXIONS). ☐ Bill me/PO# _____

☐ Charge my ☐ Visa ☐ Master Card Card # _____ Exp. Date _____

Signature _____

Please return this application with payment to:
Back issues available upon request \$10./each

CONNEXIONS

480 San Antonio Road Suite 100
Mountain View, CA 94040
415-941-3399

CONNEXIONS